



# CLOUD DATABASES AND MICROSOFT AZURE

INFO-H-415 ADVANCED DATABASES

ACADEMIC YEAR 2018-2019

BUSE OZER (000474697)

SARA DIAZ (000473952)

## TABLE OF CONTENTS

1. Abstract
2. Cloud Computing
3. Cloud Databases
  - a. DBaaS
    - i. General Explanation
    - ii. Characteristics: Advantages and disadvantages
    - iii. Architecture
    - iv. Types
      1. SQL
      2. NoSQL
4. Microsoft Azure
  - a. Introduction
  - b. Architecture
  - c. Features
  - d. Accounts and subscriptions
5. Building a database in Azure
  - a. Database used
  - b. Basic Functionalities
    - i. Creation and configuration
    - ii. Exploitation using SSMS
6. Characteristics analysis and comparison against traditional local servers
  - i. Availability
  - ii. Scalability
  - iii. Security
  - iv. Backups
7. Conclusion
8. References

## 1) ABSTRACT

With the expansion of online services throughout institutions, challenges for data storage and management have arisen. By 2010, big companies on average were producing over 65,000 petabytes (1,000 terabytes) of data each year. These massive creations of data require systems that rapidly respond to their needs.

Cloud computing-based databases offer the advantage of being scalable and dynamical to arising needs without having huge impact on the cost.

This document commences by explaining the cloud computing technology making a specific focus on its advantages and disadvantages. Having this information, a research on one of the top commercial cloud platforms in the market, *Microsoft Azure*, is done to study its characteristics (advantages and disadvantages), create comparisons with other technologies and arrive to conclusions. A real database creation and exploration is done using its services and it is captured in this document for its understanding.

Finally, a comparison against on-premises servers is done to properly understand why the cloud is becoming a trend within companies and why it is advantageous for companies (especially those of small and medium size).

## 2) CLOUD COMPUTING

Cloud computing is defined as the set Cloud computing is a model which allows the on-demand delivery of a set of resources over a network access. Often just called “the cloud” (used as a metaphor for the internet), is constituted by a web of remote servers, as opposed to local ones, that are linked together and operate as a single system. The resources provide storage, managing, use of applications, among others; and are acquired based on a pay-per-use measured basis. It was created under the idea that processing and storage could be made more efficient, scalable and elastic by using farms of data that can be accessed through rapid connections secured by a system like the internet. This has proven efficient throughout the years as it is slated to replace 14.5% of global on-premise spending in 2020 and grow from \$67B to \$162B between 2015 and 2020 (Columbus, 2017):

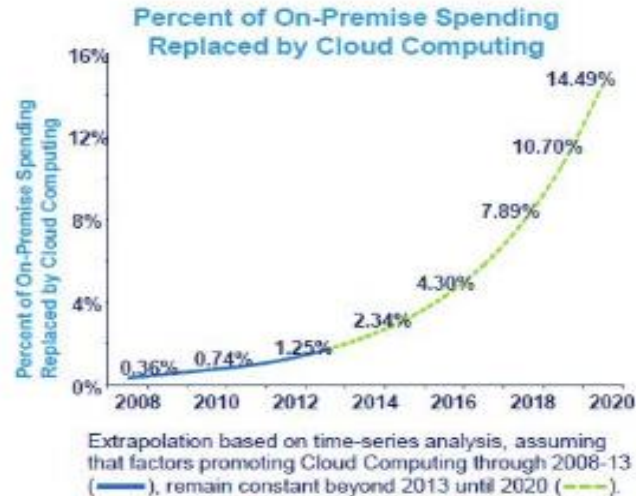


Figure 1: Columbus, L. (2017)

The cloud can be deployed under four models:(1) the public cloud in which the infrastructure is made available for the general public; (2) the private cloud whose infrastructure is reserved for a single organization; (3) the community cloud in which it is shared by several organizations with a common concern (mission, policies, etc.); and finally (4) the hybrid cloud which is a mix of the first three (public, private, community).

It currently offers four (4) service models: software as a service (SaaS), platform as a service (PaaS), Infrastructure as a Service (IaaS), which contain virtual machine images; and the most recent one Database as a Service (DBaaS).

The next figure explains summarizes all the mentioned characteristics that comprise cloud computing:

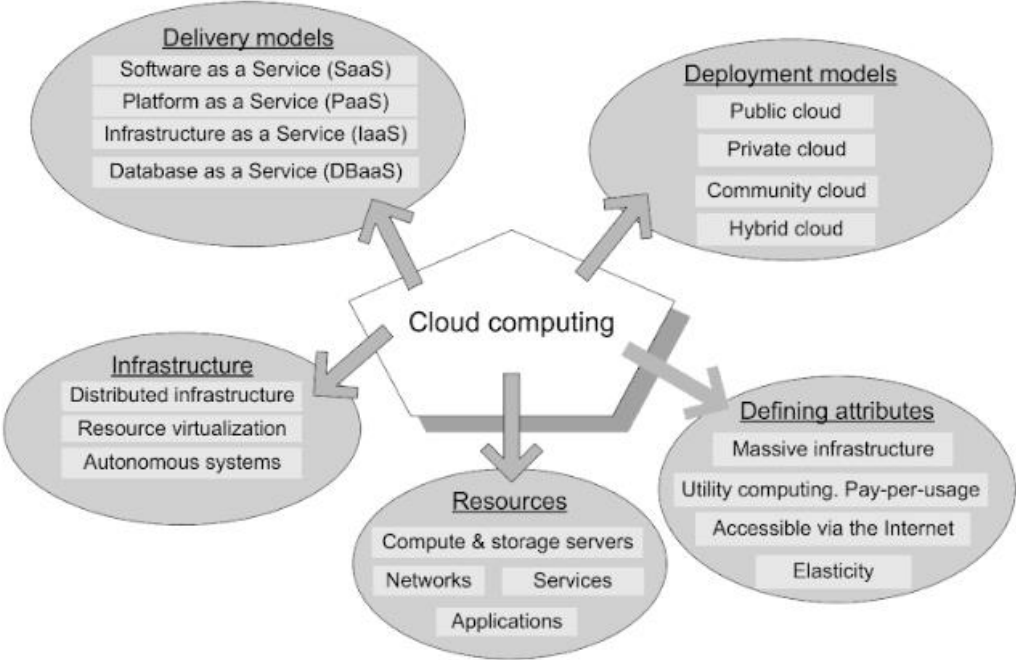


Figure 2: Cloud computing - delivery models, deployment models, defining attributes, resources, and organization of the infrastructure

As one of the main topics in cloud computing, this document will focus specifically on cloud databases, centering in one of the major industry players, Microsoft and its contributions to the field with the development of *Microsoft Azure*.

### 3) CLOUD DATABASES

A cloud database is a database that runs on a cloud computing and it allows users to store, manage, retrieve and manipulate their data via a cloud platform, accessible over the internet anytime. The type of data can be structured, unstructured and semi-structured. Cloud databases are responsible for availability and high scalability which means they take care of being up all time and be able to manage a data load.

In addition, the usage of cloud database has been growing rapidly over time. There are many cloud database providers in the market. Therefore, when it comes to choose a product among all of them, it needs to be considered carefully. AWS, Microsoft Azure, IBM Cloud, Google Cloud, Oracle Cloud, SAP Cloud are the key players of cloud databases in the industry. Currently Amazon has the biggest shares in the market however it doesn't mean that Amazon will have the majority in the future. Moreover, the capital expenditures growth rate of other companies has decreased slightly but Microsoft will still spend more money than it used to in order to keep up with its competitors like AWS and Google and it has increased its revenue up 87% over the past year.

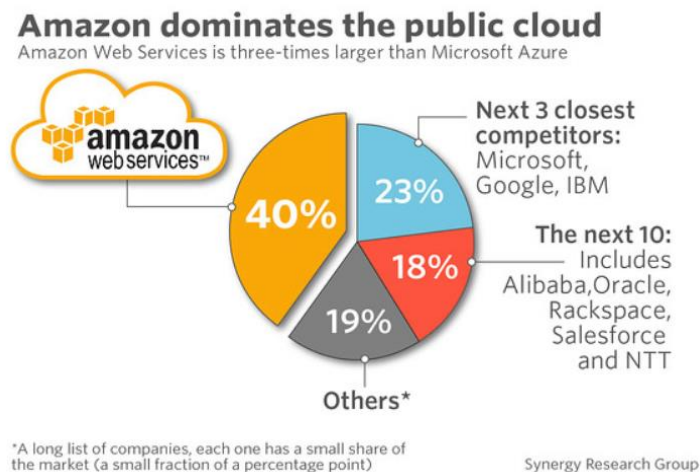


Figure 3: Garcia, T. (2017)

Cloud vendor	Annual revenue run rate
Microsoft commercial cloud	\$21.2 billion
Amazon Web Services	\$20.4 billion
IBM	\$10.3 billion
Oracle	\$6.08 billion
Google Cloud Platform/G Suite	\$4 billion
Alibaba	\$2.2 billion

*Source: Company filings, earnings reports*

Figure 4: Annual revenue per cloud vendor in 2017

Cloud database is an optimal choice in order to develop a database system that is dynamically scalable depending on the user needs because they have pay as you go policy based on usage of cloud database. User can scale up and down as needed. It also helps to reduce the cost for developing and managing a database because traditional databases should be managed by DBAs. Installation, configuration, upgrading of a related database, database design and implementation, database backups, creation of user accounts and data transformation are the responsibilities of DBAs. All the above-mentioned steps are overhead for most companies because it requires more workload, money and a reasonable amount of time.

“As the cloud continues to go drive the cost of storage and compute, a new generation of applications have emerged, creating a new set of requirements for databases. These applications need databases to store terabytes to petabytes of new types of data, provide access to the data with millisecond latency, process millions of requests per second, and scale to support millions of users anywhere in the world. To support these requirements, it is needed both relational and non-relational databases that are purpose-built to handle the specific needs of applications”, Jeff Bezos, the founder of AWS. To conclude, cloud platforms bring us flexibility, convenience and a variety of possibilities.

## a) Databases as a Service (DBaaS)

- General Explanation

DBaaS is a database service which is managed and deployed in the outside provider's infrastructure. According to some authors, it seats between SaaS (Software as a Service) and Platform as a Service (PaaS):

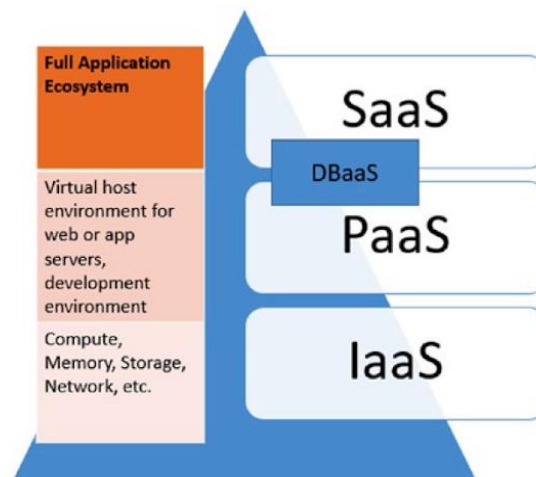


Figure 5: Location of DBaaS between SaaS and PaaS

SaaS offers the possibility to choose already constructed applications developed by the provider while PaaS gives the tools and programming languages to deploy those created by the user. DBaaS is a service that is a hybrid of both as it works by giving the user the option to choose the database technology to deploy he or she wants to use, much as it is in PaaS, but the user won't be doing much beyond choosing some specific features such as sizing which is pretty much how applications work on SaaS.

- Characteristics

DbaaS characterizes itself on the next number of advantages:

- Self-service: easily configurable without the need of complicated deployment or configuration



- On-demand model: More controlled (and probably lower) IT costs as the user pays metered according to the needs
- Elasticity and scalability: It gives a dynamic, flexible and even automatic way to scale up or down according to requirements
- Increased accessibility: Due to the fact of it being deployed on a remote network, it offers access from many locations at any time
- IT Expertise: The service comes with a support of specialists who know the system and how to handle new requirements or inconveniences. This also helps free the internal organization's personnel to perform other tasks
- Agility: Applications adapt easily to the introduction of new technologies which are constantly being updated giving the latest versions
- Automated technologies: The service has a series of automated processes in case of failure such as automatic failover, recovery from failure and auto-scaling
- Multi-tenancy: Running the same instance of software within a cloud server to server many tenants allows the reduction of deployment and development costs.

However, even with its rapid growth through the industry, it still has disadvantages that must be considered before its acquisition:

- Control and privacy: Even if having the support of experts is a relief, it also means losing control of the underlying systems supporting the databases which might be a risk in terms of security as it can be lost or end in the wrong hands.
- Size vs. cost: When companies reach a certain size, like mega corporations as Google, depending on someone else might be way too expensive to rely on outsourcing its databases.
- Downtime: The connection between the remote servers affects directly the performance of the database. This is a problem that might be more easily solved with an in-house solution but with a remote connection might require an updated and sound communication technology.

- Vendor Lock-in: differences between vendor's platforms might create difficulty on migration between products which results on un-flexibility and higher costs when a change is desired
- Architecture

Cloud based storage environments have different architecture alternatives that can be used by DBaaS providers according to client's needs. For management of cloud-based databases a newly architected DBMS was required to fit the requirements of Cloud computing platforms. An approach for this needed design can be seen in the layered architecture proposed by Y.E. Gelogo :

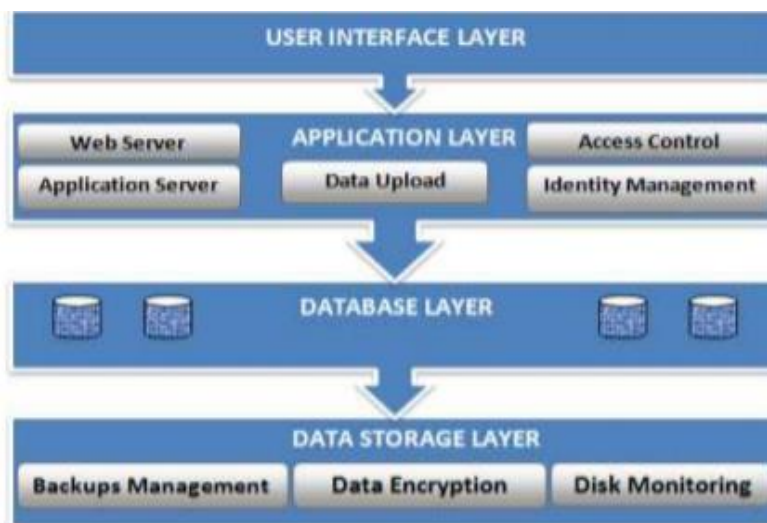


Figure 6: Layered Architecture proposed (Gelogo., 2012)

User Interface Layer to access the service via internet. The Application layer is used to access software services and storage space on the Cloud. The Database layer provides efficient and reliable service of managing database by reusing query statements residing in the storage, thus saving time for querying and loading data.

Data is encrypted when stored or backed up at Data storage layer, without any need for programming to encrypt and decrypt. Other architectures used by cloud services are:

- Shared memory architecture: many CPUs access the memory so that they can communicate
- Shared disk architecture: disks are shared by processors to act as a network.
- Shared nothing architecture: each service has a respective database, and it is accessed by only that service and there is no disks or memory shared.

- Types

- i. SQL

SQL is a standard language for relational database management systems that allows to communicate with a database in order to manipulate and retrieve data. It is widely used in the market for many years because it has a standard language, easy to use and optimized for general problems. Although it allows great flexibility in many cases, it leads to inconvenience to find appropriate solutions for special cases because of the generalization. In addition, the data is divided into groups, referred to as tables so it is called table-based. To begin with SQL, predefined schemas are required in order to determine the structure of data before populating a database. Once schema has been implemented, it can be no longer changed. Therefore, before implementing schema, development process takes a reasonable amount of time. In addition, SQL databases only support structured data and all of data must follow the same structure. It performs well for complex queries however it has some restriction because of the same structure policy. Moreover, SQL databases are vertically scalable. It means that in case of increase in load, it can be solved by increasing CPU, RAM, SSD on a single server which is costly and undesired.

MySQL, Oracle, SQL, PostgreSQL can be given as an example of NoSQL databases.

Relational SQL databases can be chosen for following scenarios:

- pre-defined schemas
- multiple-row transactions
- critical data

- small size of data
- low complexity

## ii. NoSQL

The widespread use of internet and increase of large web applications leads to complex data structures and increase in data volume. Therefore, traditional relational databases are no longer maintain the data for this reason key players of the industry started to search new solutions to be able keep up with the evolution of technology. An alternative to SQL, NoSQL has emerged over time. NoSQL started with the idea of NoSQL which means that I do not want to use SQL. However, the meaning of the terminology turns out not only SQL.

*“NoSQL is a generic term used to refer to any data store that does not follow the traditional RDBMS model, specifically data is non-relational, and it does not use SQL as a query language. It is used to refer to the databases that attempt to solve the problems of scalability and availability against that of atomicity or consistency.”* Gaurav Vaish, Getting started with NoSQL.

In addition, there are a several number of features that NoSQL has. First of all, NoSQL performs on unstructured data and does not require data that follows the same structure. They can be in a different format. Unlike SQL, there is no need to predefined schema, which is called schema-less data representation, before populating a database. which prevents us thinking ahead and wasting time in the development process. It can be started directly by populating a database. This enables to have a dynamic schema which can be changed anytime. Because the data is not relational, there is no longer need to join tables which enables us to have better performance. Lastly, one of the most important features that NoSQL solves is better scaling according to needs. Unlike SQL, NoSQL is horizontally scalable. This means that more traffic can be handled by adding more server and sharing. Objects in NoSQL databases can be resided across different servers because it only promises a record-level atomicity rather than a series of transactions. The main reason of difficult implementation of scaling in a SQL database is that it guarantees ACID properties. However, NoSQL provides

BASE properties which are basic availability, soft state and eventual consistency. This enables us to scale out easily and efficiently. According to CAP theorem, it is impossible to have 3 properties which are consistency, availability and partition tolerance at the same time in distributed systems. Only two out of three can be provided.

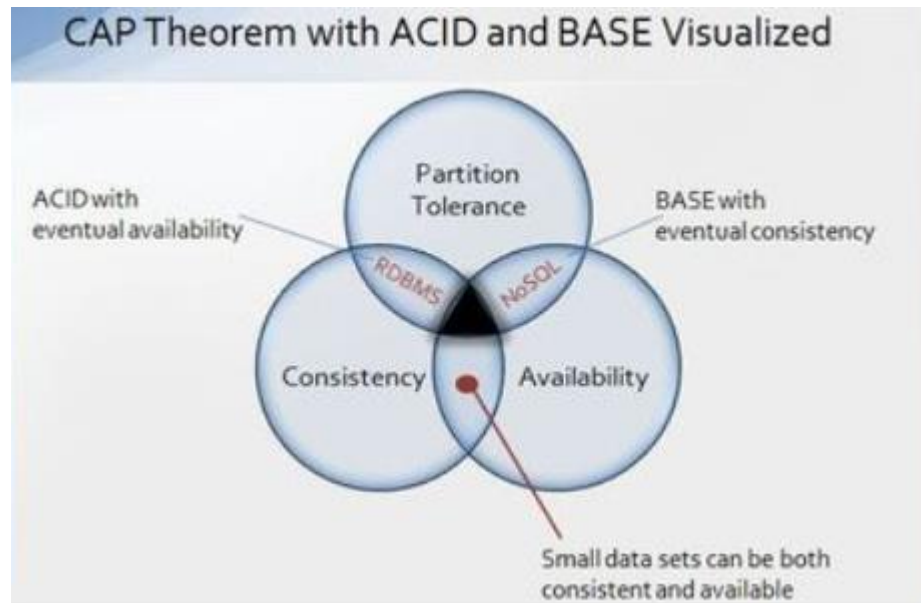


Figure 7: CAP Theorem

Although there are several advantages to choose NoSQL, there is no standard for NoSQL, a language varies database to database which makes it hard to switch databases and a limited number of people know different specific NoSQL databases.

MongoDB, BigTable, Redis, RavenDB Cassandra, HBase, Neo4j and CouchDB can be given as an example of NoSQL databases.

Non-relational NoSQL databases can be chosen for following scenarios:

- a rapid growing business
- no clear schema definition
- continuous change
- large volumes of data

## 4) Microsoft AZURE

### a) Introduction

Azure is a private and public cloud platform that offers a set of services all developed by Microsoft Inc. These services help IT specialist to build, deploy and manage applications. Its most popular products include:

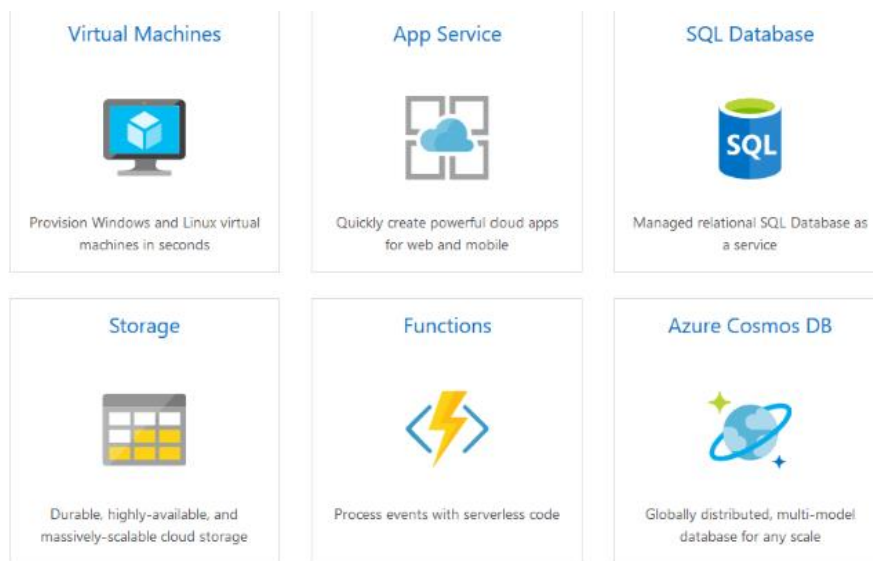


Figure 8: Azure feature products

Currently, Azure counts with more than 100 products each offering solutions like:

- Artificial Intelligence
- Internet of Things
- Blockchain
- SAP on Azure
- DevOps
- Hybrid Cloud applications

By 2018, Azure counts with 54 regions to host its operations:



Figure 9: Azure datacenter presence worldwide

Database products like SQL Azure DB are present in 46 of these locations (85% of the total). This gives clients the opportunity to choose a desired location according to their needs of response and latency times, as well as IT knowledge in the region, risk of natural disaster occurrence, etc.

## b) Architecture and Design

Azure uses a technology called Virtualization which separates the coupling between a computer's hardware and operating system by using a layer named a **Hypervisor**, which emulates a computer's functions and CPU within virtual machines. These virtual machines can run both Windows and Linux operating systems and optimize the abstracted hardware.

Each of its data centers distributed around the world contains many racks which containing servers and each of this implements this Hypervisor technology to run multiple virtual machines. A network switch provides connectivity between the servers. create it within the servers. After this the user can access it and all its functions. Each server within the rack runs a special piece of software called the Fabric Controller, which, at the same time, is connected to another software called the Orchestrator which manages every activity in Azure included, but not limited, to user requests. These requests are created by using the web API, which is more popularly accedes through the Azure Portal (there are other ways to access it). The request is packaged and sent to the Fabric Controller:

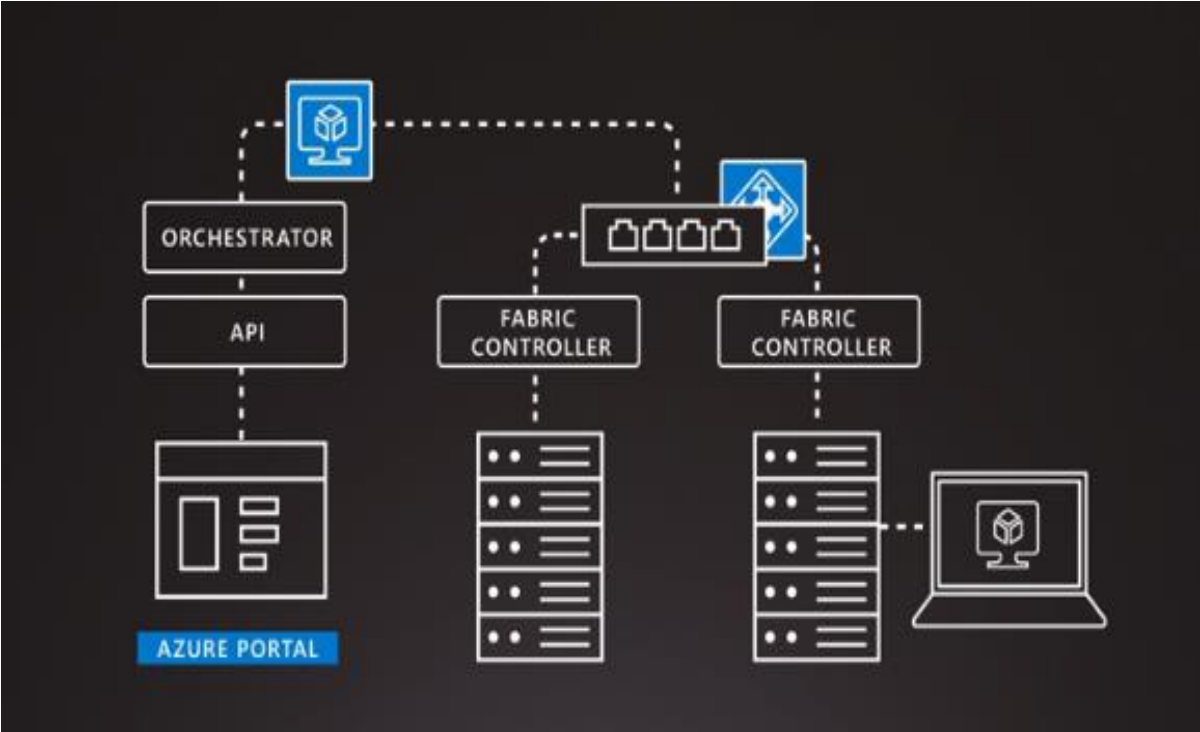


Figure 10: Azure portal's connection to data centers

For its relational cloud databases, Azure, as explained in the previous part of this document, uses a layered architecture:



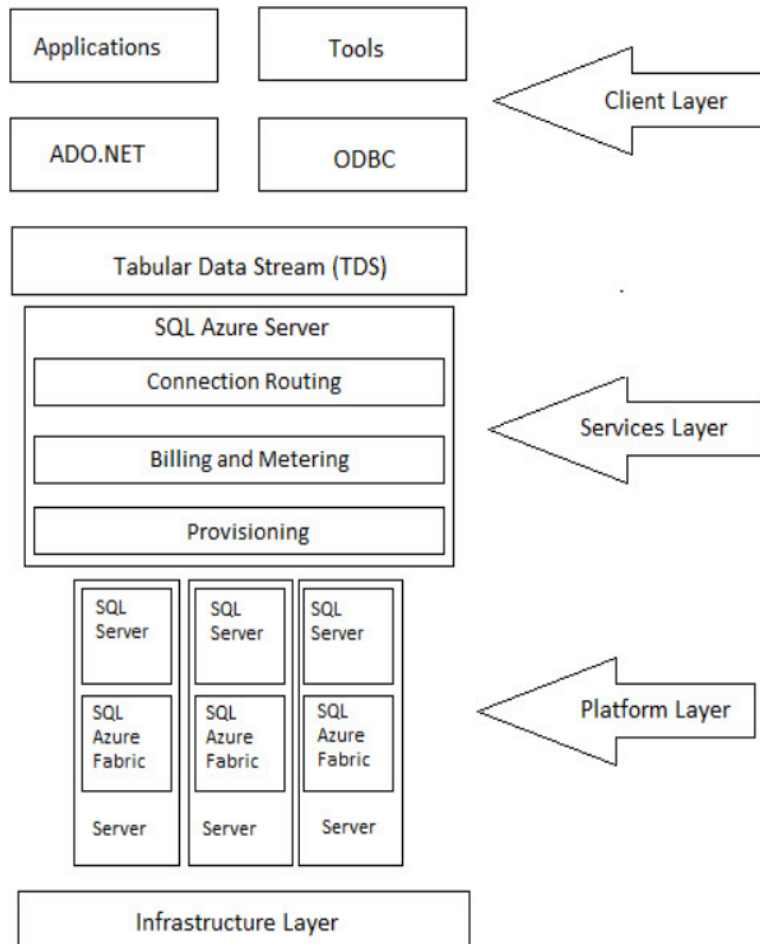


Figure 11: Azure layered architecture

- Client Layer: serves as the connection between the users and Azure. This layer uses TDS Interfaces.
- Services Layer: As the connection between the client and platform it is responsible for creating and provisioning the databases created by the Azure platform of SQL Server Management System. It also controls the routing connections between physical servers and applications.
- Platform Layer: Manages all of the server instances and services. This is controlled by the SQL Azure Fabric controller which provides computing balancing, automatic failover and replication between the different servers.

## c) Features

Azure divides its services into two main groups: infrastructure and Platform Services:

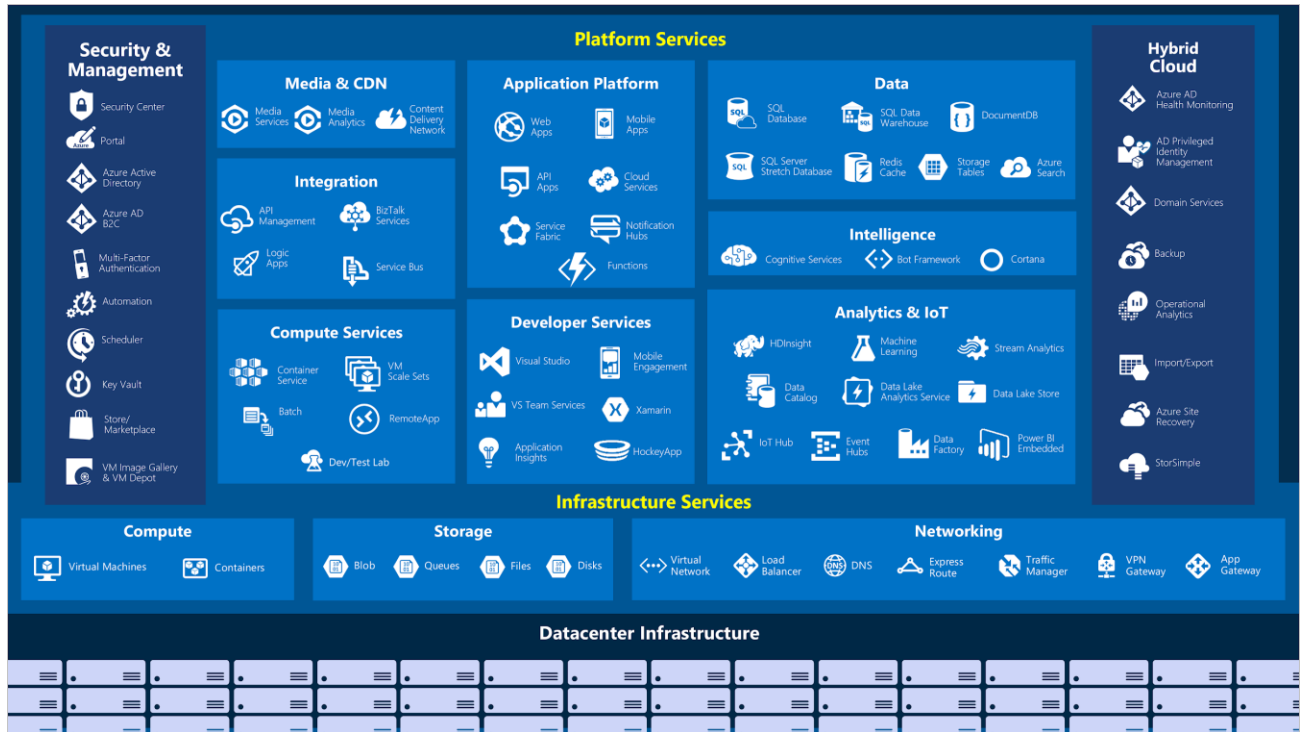


Figure 12: Azure service map

The platform services are also supported by Security and Management services as well as they can be deployed on hybrid cloud services (public and private clouds).

The most commonly used features (which encompass numerous of these services) are:

- **Compute:** Applications and services which offer calculation, logic execution and running applications capabilities. This include virtual machines, serverless computing, etc.
- **Networking:** Mainly controlling and configuring of traffic into and out of Azure as well as more specialized features like private connections to on-premises environments
- **Storage:** Storage for different data types

- Mobile: App development and other enterprise adapted services as offline data synchronization, connectivity to on-premises data, etc.
- Databases: Azure provides multiple database services that contain variety of data types and volumes:
  - SQL Databases: Fully managed relational databases which use SQL querying language and are equipped with intelligence and security.
  - Azure Cosmos DB: A globally distributed multi-model database, which supports NoSQL
  - Azure Database for MySQL: A fully managed MySQL relational database
  - Azure Database for PostgreSQL: A fully managed PostgreSQL relational database
  - SQL Server on Virtual Machines: To host enterprise SQL Server apps in the cloud
  - SQL Data Warehouse: A managed, elastic data warehouse with security at every level of scale at no extra cost
  - Azure Database Migration Service: Migration to the cloud with no app code changes
  - Azure Cache for Redis: This product offers superior throughput and latency performance by storing data in memory instead of on disk
  - Table Storage: A NoSQL key-value store for accelerating development by using semi-structured datasets
  - Azure Database for MariaDB: A fully managed MariaDB relational database
- Web: web applications and services such as search engines, API online management, etc.

#### d) Accounts and subscriptions

To create an Azure account all that is needed is an e-mail address and billing information. Every account is associated with one or more subscriptions. Each subscription holds defined resources and services. The client will be billed according to each subscription.

Subscription types:

- Azure free subscription: A trial for clients that offers a US \$200 credit
- Azure Student subscription: the same idea as the free trial but it's directed to students, it lasts for 12 months and US \$100 credit is given
- Azure Pay-As-You-Go: Monthly billing for the services used. Serves a large range of clients from small to massive companies
- Azure Enterprise Agreement: provides flexibility for enterprise level clients to get software licenses and services with included discounts

Additional to these subscriptions, Azure also offers the possibility of having **tenants**. An Azure AD tenant is a modern identity provider that supports multiple authentication protocols to secure applications and services in the cloud. It is an isolated instance of the Azure Active Directory owned by an "organization" that could be a team, a department, a company or other group of people. This allows the multi-tenancy characteristic of cloud services.

## 5) Building a database in AZURE

### a) Databases used

For the creation of a test database, a simple relational model of three (3) tables was used and acquired from:

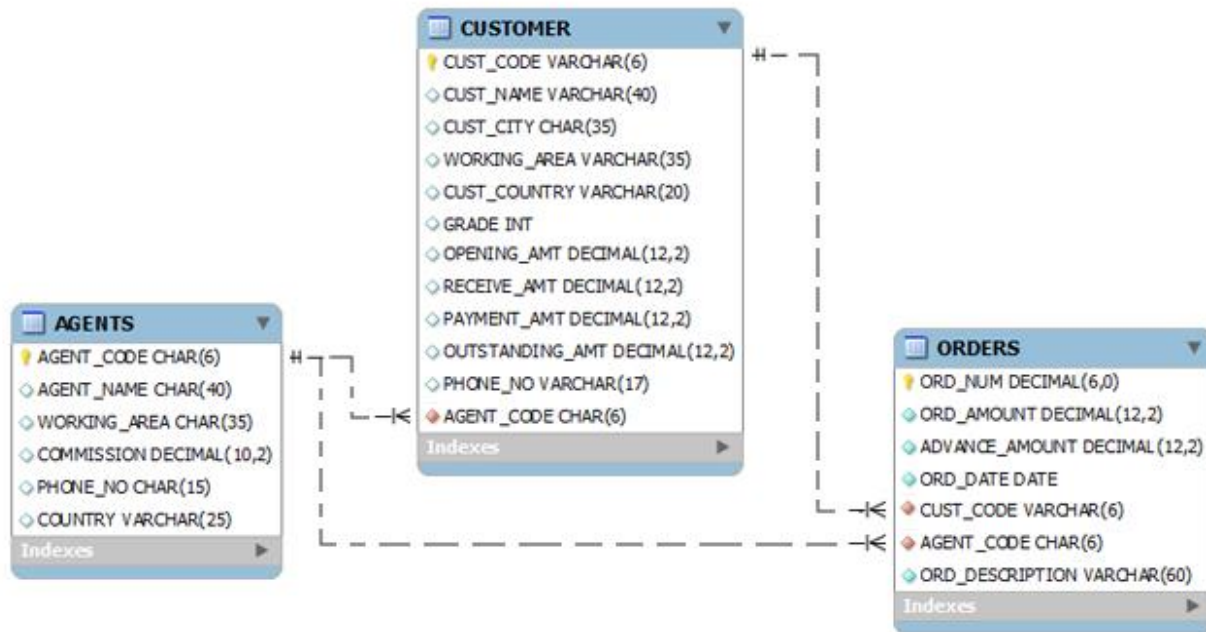


Figure 13: Conceptual diagram of the database used

### b) Basic Functionalities

For the purpose of this document an Azure student license was acquired, giving the team the possibility to use and explore some of the features that the platform has to offer as it was limited by the amount of credits given (US \$100).

- Creation and configuration

Enter <https://portal.azure.com/signin/> and sign with the given account credentials.

Chose *Create Azure service* and then chose to create a database:

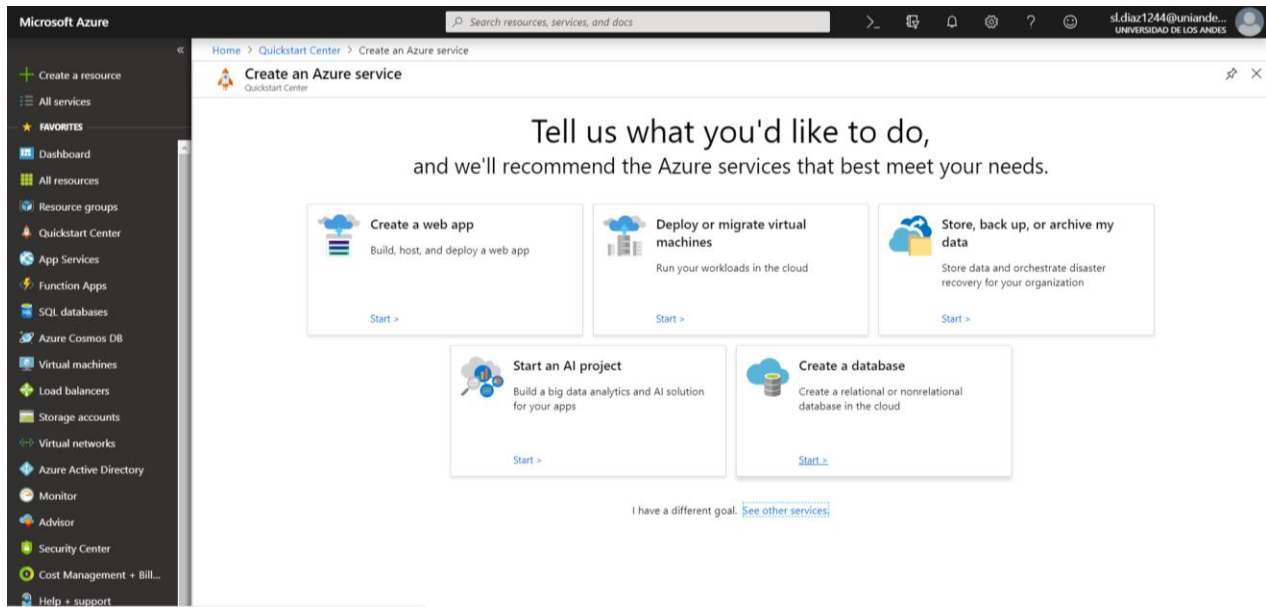


Figure 14: Building a db in the Azure platform

Two options are available:

- Azure SQL Database: a relational database
- Azure Cosmos DB: globally distributed, multi-model database service that supports document, key-value, wide-column, and graph databases. This type of database is a good choice for “new web, mobile, gaming, and IoT applications where automatic scale, predictable performance, fast order of millisecond response times, and the ability to query over schema-free data is important.” This means that it is more suited for non-schema databases, which is not the case with the test case chosen.

So, we chose an Azure SQL Database:

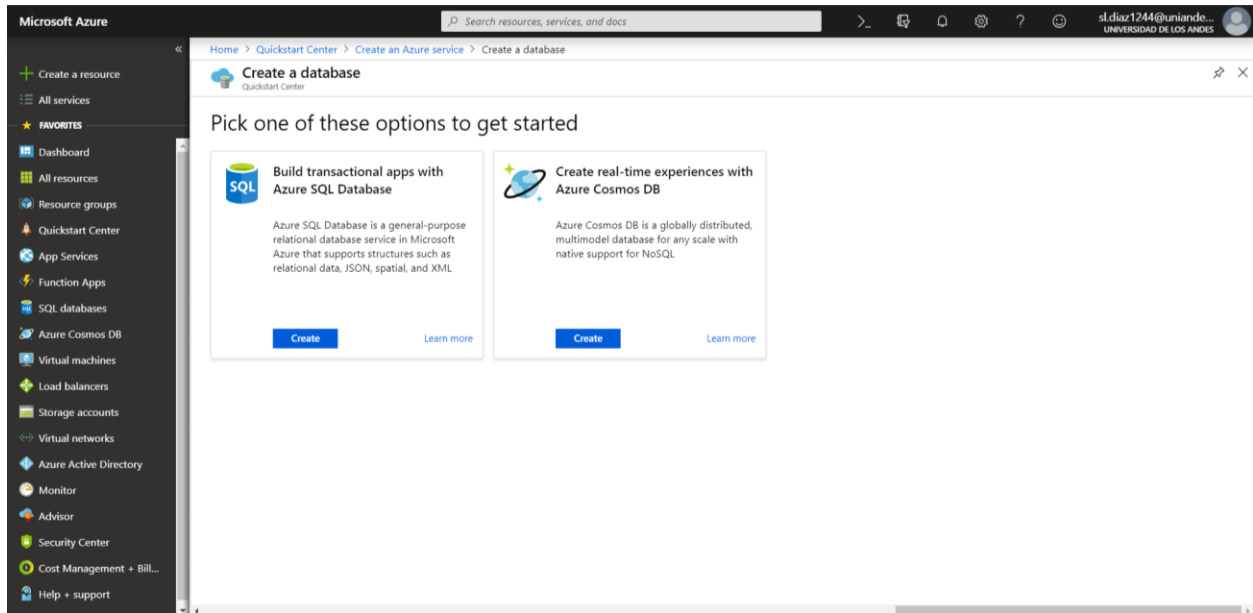


Figure 15: Building a db in the Azure platform

The parameters needed to configure it are:

- Database name: add a name of your choice, we named it “DB”
- Subscription: Azure subscription type, in this case will be the student one
- Resource group: groups a set of assets in logical groups for easy or even automatic provisioning, monitoring, and access control, and for more effective management of their costs
- Source: blank database, sample or a backup
- New Server:
  - 1) Server name: add a name of your choice, in this case we named it `adbproject.database.windows.net`
  - 2) Server admin login: set a login
  - 3) Server admin password: set a password
  - 4) Location: chose the location of the server, in this case we chose France Central as it is the closest to our location which might make the connection faster and for security reasons it's also closer

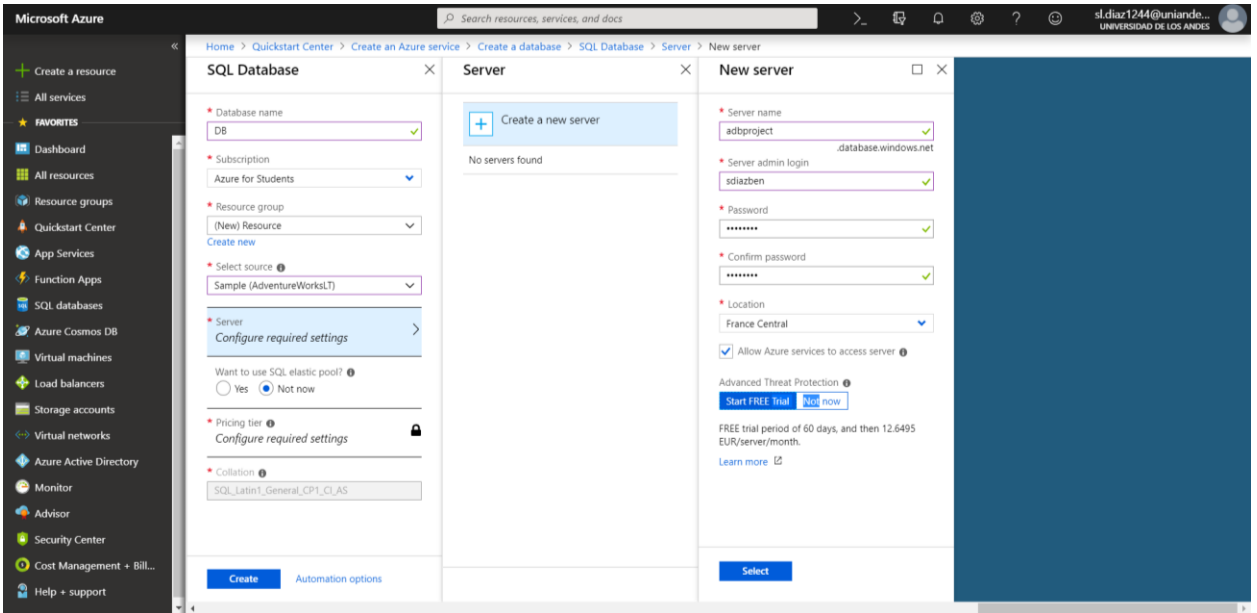


Figure 16: Building a server in the Azure platform

- Elastic pool: is a shared resource model that enables higher resource utilization efficiency, with all the databases within an elastic pool sharing predefined resources within the same pool. This has an hourly additional cost.
- Pricing tier: to specify the service tier, the number of DTUs (Data Transaction Unit: a way to describe the relative capacity) and the amount of storage. The pricing of this could be:

Basic	Standard	Premium
For less demanding workloads	For workloads with typical performance requirements	For IO-intensive workloads.
Starting at 5.25 EUR / month	Starting at 15.82 EUR / month	Starting at 490.17 EUR / month

Figure 17: Azure servers' price tiers

It may also go up depending on the amount of DTUs. We chose the Standard Configuration as no extra storage is needed (10 DTUs and 250 GB):



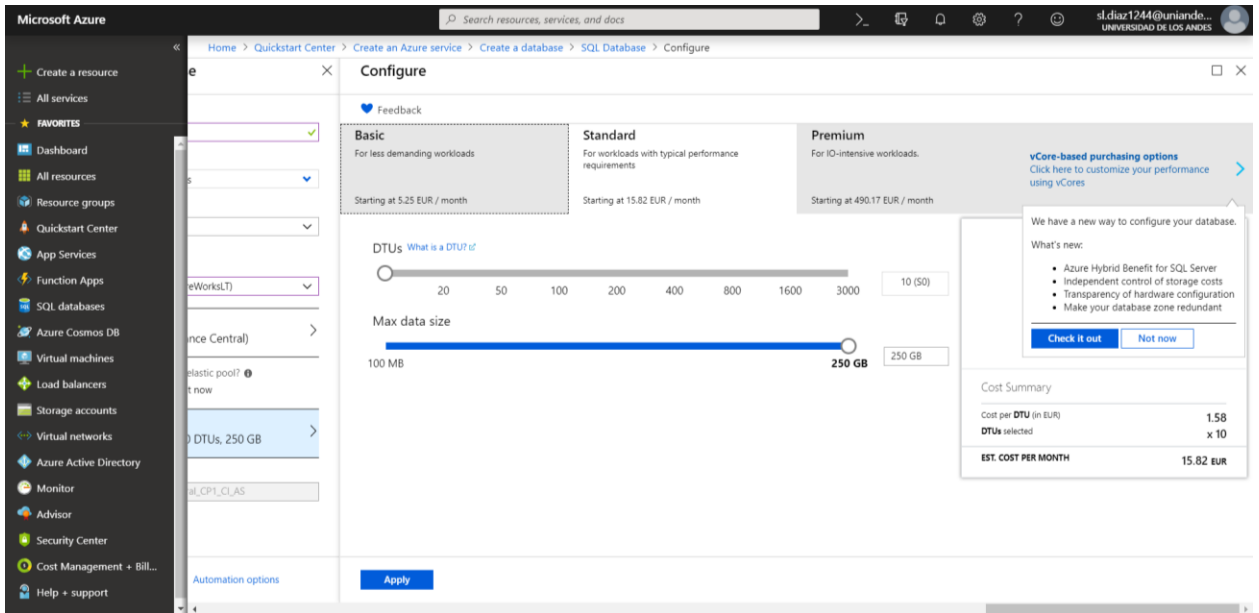


Figure 18: Azure servers' price tiers

→ Collation: set SQL\_Latin1\_General\_CP1\_CI\_AS

With this, we successfully create the database, and then when we chose our databases on the left menu, we can choose it and overview its performance.

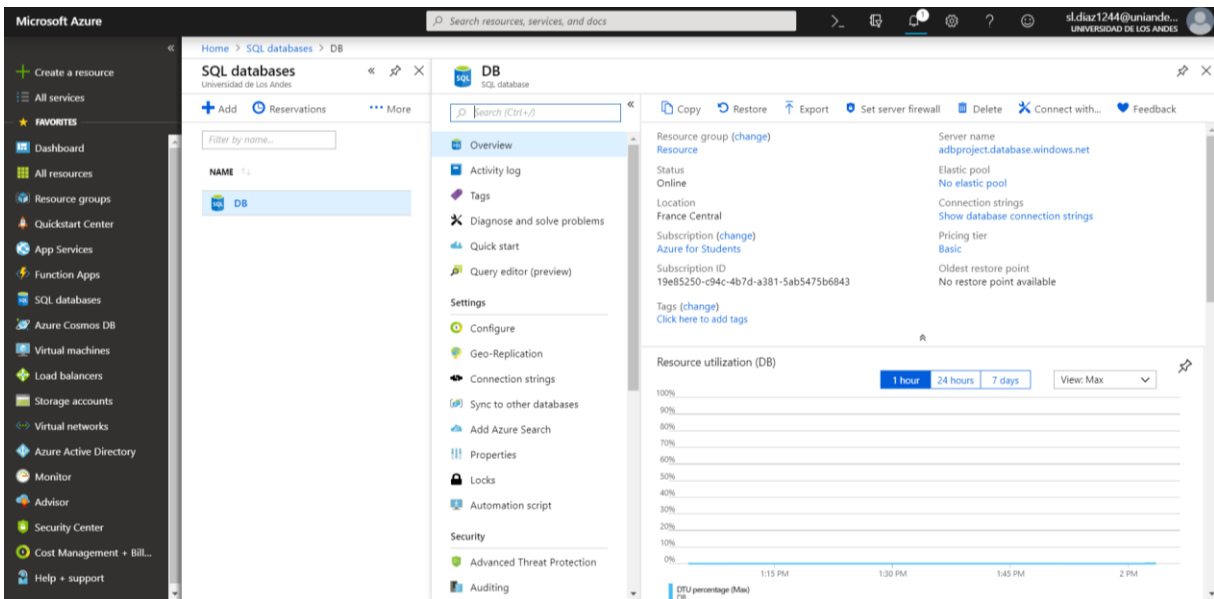


Figure 19: Building a db in the Azure platform

- Exploitation using SSMS:

Once we have created the server and the database, we can use an integrated environment as SSMS (SQL Server Management Studio). Here we just simply add the name of the server and the respective credentials:

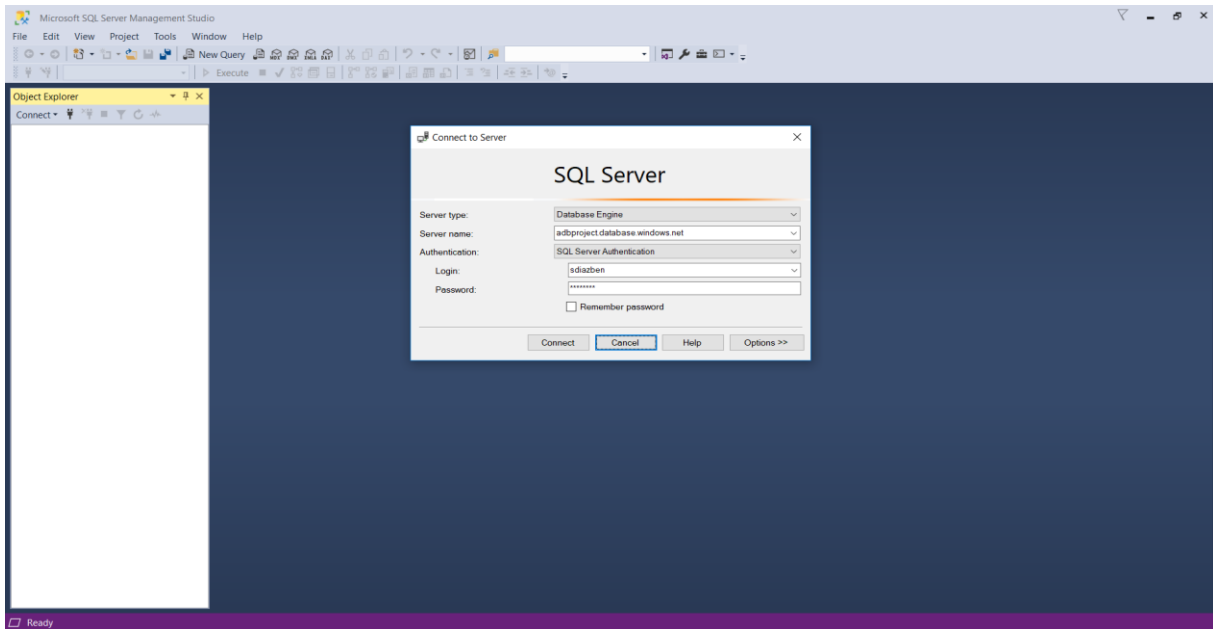


Figure 20: SQL Server exploitation

We log-in into the Azure account:

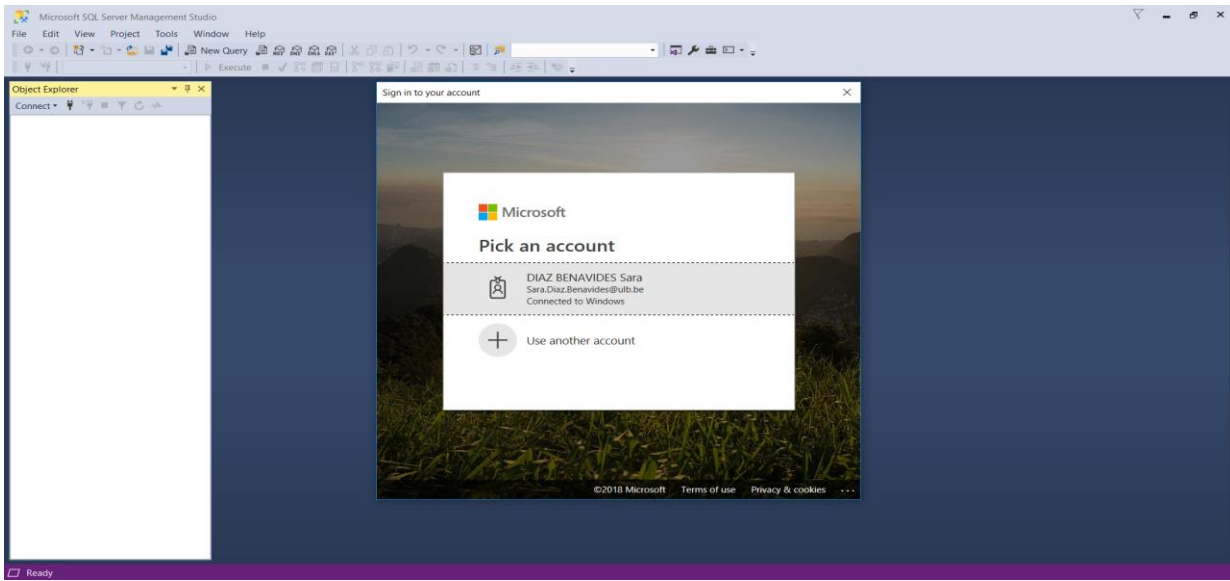


Figure 21: SQL Server exploitation

And finally, we authorize and add the new client IP address:

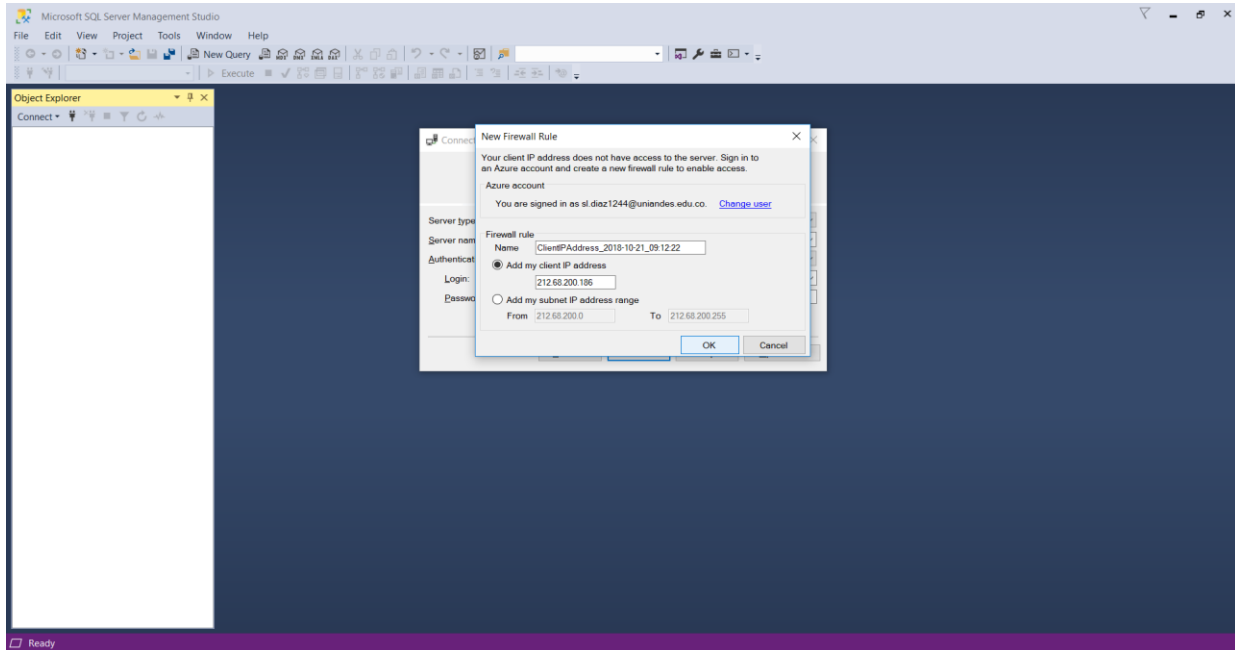


Figure 22: SQL Server exploitation

We are set to start exploiting the database (in this we named it DB):

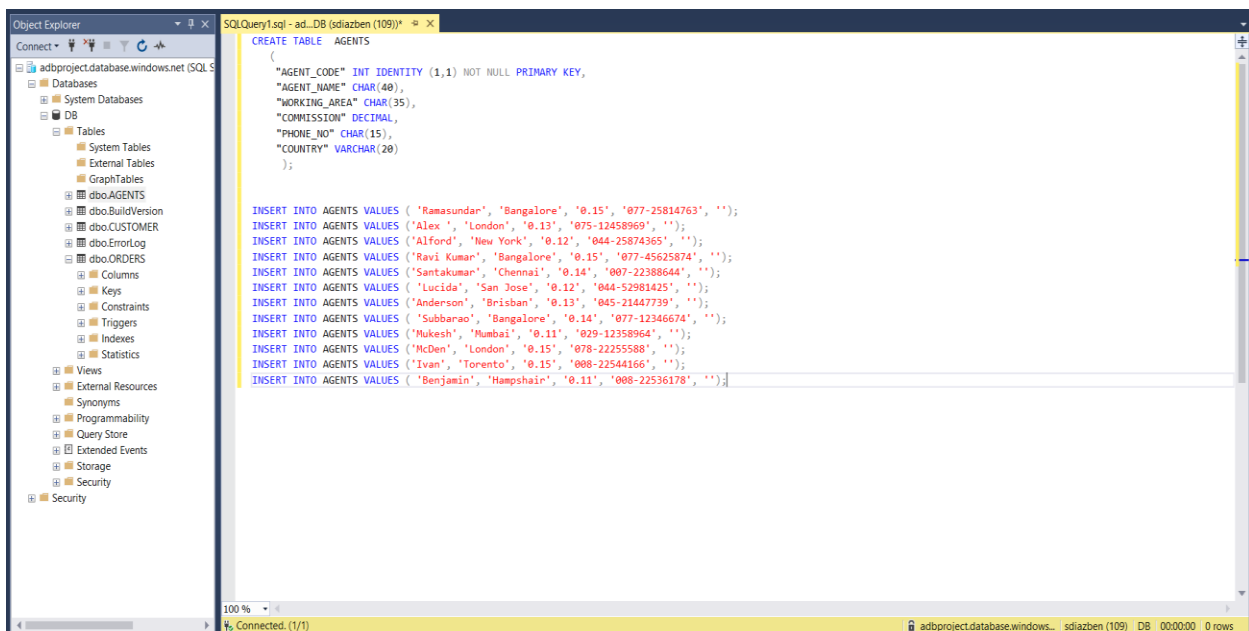


Figure 23: SQL Server exploitation

As it can be seen, SQL is used as the querying language (and extensions like T-SQL) can be used as in a local server. In this case, we inserted the data making use of a SQL script like it would have been done in a local server.

## 6) CHARACTERISTICS ANALYSIS AND COMPARISON AGAINST TRADITIONAL LOCAL SERVERS

### a) Availability

One of the most important differences between stand-alone SQL Server and Azure SQL DB is high level of availability which is achieved by running multiple number of instances on different machines. As a result of this, a few numbers of dedicated machines should be rented and managed always availability groups, created SQL Server failover cluster and monitored. It leads to higher cost and performance penalty. Although it is hard to maintain a high availability all time in local SQL Server, Microsoft Azure ensured being up and running 99% of time without extra configuration, replication, or database costs. It automatically takes care of failure detection, software or network failures. In addition, Azure SQL Database is based on SQL Server Database Engine architecture that is adapted for cloud environment to guarantees 99.99% availability even in case of infrastructure failures. Also, this feature comes automatically, and it is available for all editions of Azure SQL.

When it is wanted to achieve high availability in SQL server, the total cost is composite of server hardware, space, power and cost of administration. All costs are monthly calculated [8]. Regular server costs with 32 GB storage \$ 270 on average, power consumption cost \$ 27, space costs \$ 10 monthly according to current market prices. Also, it is required to duplicate the product for other instances and make a workload test. It takes effort and time. As a conclusion, at the end of the day, your server in a data center costs \$ 307 in total excluding data administration cost. Also hiring a data administrator on average costs US \$5000. However, if it is selected one of the basic options of Azure SQL Database, it costs 1,424 \$ per month but high availability feature comes automatically with it, there is no additional cost.

**Azure SQL Database**

REGION: East US | TYPE: Managed Instance | BACKUP STORAGE TIER: LRS

SERVICE TIER: General Purpose | GENERATION: Gen 4

INSTANCE: 8 vCore

---

**Billing Option**

Save up to 73% on pay as you go prices with 1 year or 3 year reserved options.

- Pay as you go
- 1 year reserved (~21% discount)
- 3 year reserved (~33% discount)

Save up to 55% with [Azure Hybrid Benefit](#) for SQL Server

1 Instances × 730 Hours = \$1,472.75 Per month

Figure 24: Azure SQL DB calculator

## b) Scalability

Azure SQL Database is a service database that can be easily scaled up and down based on needs. More compute or storage resources can be adjusted to meet your system requirements without buying new hardware or migrating data to more powerful machines. Microsoft Azure SQL enables to change performance characteristics of a database on the fly and assign more resources when it is needed in case of heavy load. It releases resources when they are not needed in order to decrease the cost. It allows tailor a database performance to satisfy workload demand.

Azure SQL Database supports both vertical and horizontal scaling.

The necessity of scaling up comes into play when there are performance issues that cannot be solved by classic database optimization techniques. This type of problems should be fixed as quickly as possible when a current performance level cannot handle all requests. More

resources such as CPU or IO can be added in case of peak load, then resources can be released when there is no need. This is an example of vertical scaling. It is highly flexible platform based on pay-as-you-go.

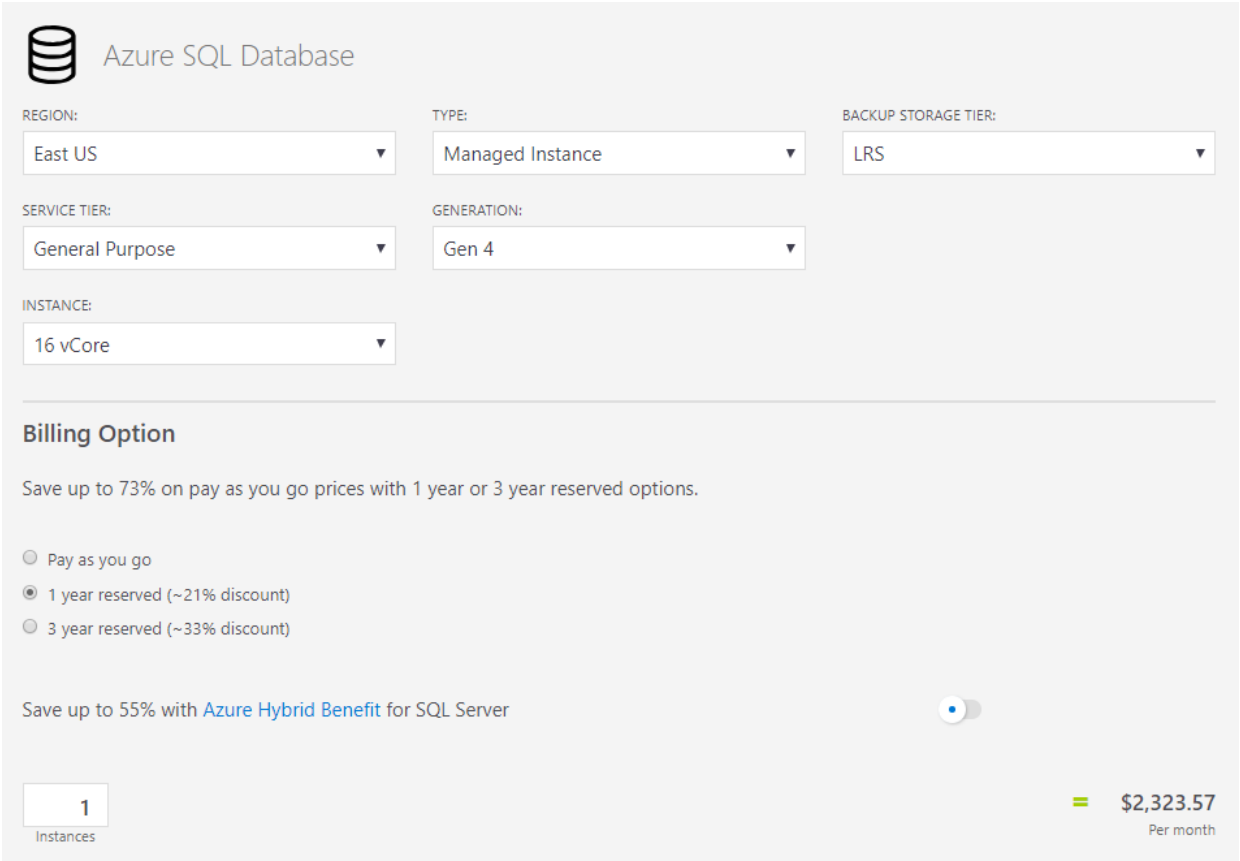


Figure 25: Azure SQL DB calculator

If it does not help to solve performance issue, even though when more resources are adjusted, it is a good idea to consider horizontal scaling which allows to split the data across several databases and scale up/down independently.

Microsoft Azure enables to choose how many CPU and how much storage is needed and dynamically change parameters for resources such as CPU and I/O any time.

In addition, it can be selected another model based on DTU (database transaction units) which represents a pre-defined bundles of computer resources such as of CPU, memory and I/O and included storage to drive different levels of application performance. It might be

convenient to customers who need pre-configured model without overhead of defining parameters. Also, the payment is fixed each month, unless DTU bounding box access its limit, it is always used the defined DTU class without scaling up a database. If DTU bounding box reaches its limit, it goes with a higher DTU bounding box to handle the workload.

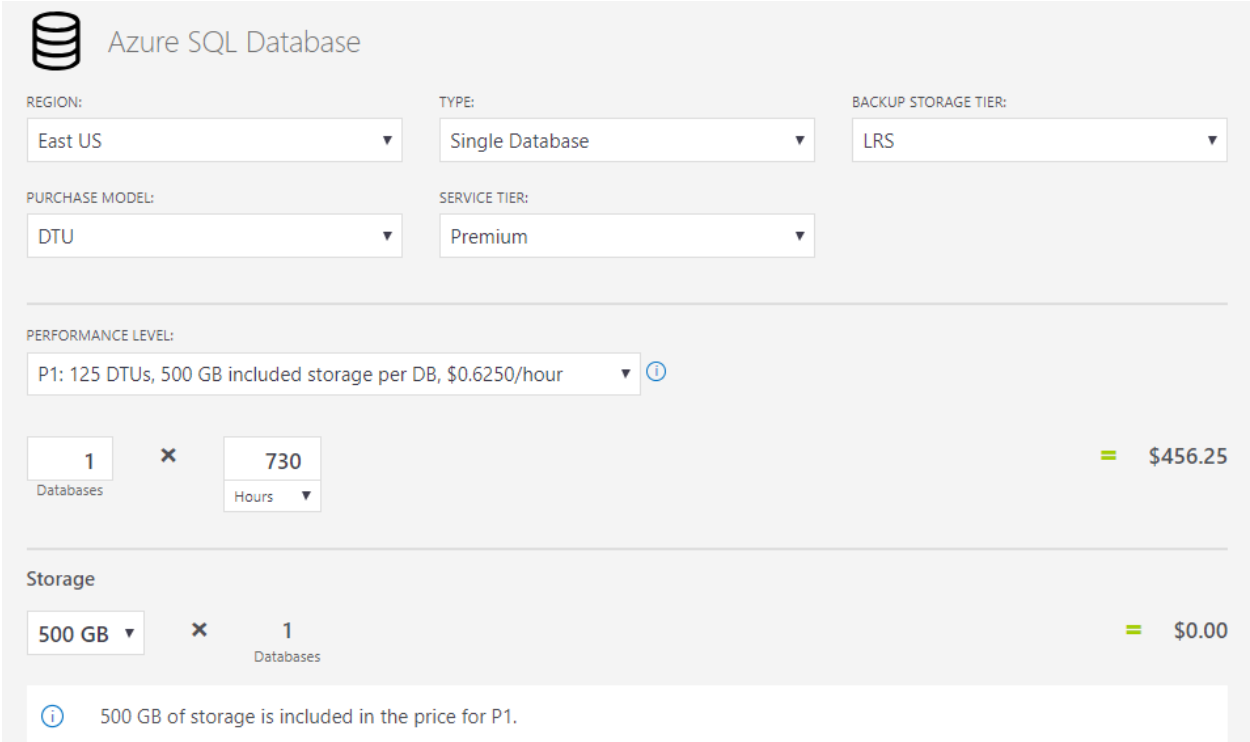


Figure 26: Azure SQL DB calculator

Although If we want to upgrade our stand-alone server from 64GB of RAM and 8 CPU to 512GB of RAM and 32 CPUs, it costs \$9,895 according to [9], Azure costs \$8,796.

When scaling up cannot be done, there is another option that might be considered is redirecting part of workload to another database node. There are several replicas provided by high-availability feature. They become active when a primary node fails. These read-only replicas can be used for Read Scale-Out feature which allows to perform read-only queries. Therefore, it can be accomplished by dividing the workload into two groups which are read-only and read-write. Read-only replicas can be used purposes such as analytics. This feature enables us using the additional capacity without paying extra cost. The benefits of read scale-

out model is that primary node does not spend resources for read-only queries anymore. All read-only / analytic queries send to secondary node. It leads to improving performance for read-write queries. Also, secondary node takes care of all read-only / analytic queries which includes heavy loads.

### c) Security

One of the most fundamental differences between stand-alone SQL Server and Azure SQL DB is security. Security is a top concern for managing databases for many years. Even though most people say that it is risky and dangerous to store data on the cloud, the way of implementation is imperative in order to keep the data secure. If it cannot be provided a quality implementation, it is a good way to choose Azure SQL Database. It provides built-in security features that help to protect your databases. One of the most important features is encryption that is a way of Azure securing data in motion with Transport Layer Security and for data in use Always Encrypted. Azure SQL Database enforces SSL/TLS encryption for all connections, which guarantees all data is encrypted "in transit" between the database and the client. Second feature is data discovery and classification such as threat detection which detects anomalous, unusual activities and potentially harmful attempts to access database. Also, this feature allows you to control access your highly sensitive data, to help meeting data privacy standards and compliance regulatory requirements. The way of controlling access is achieved by authentication mechanisms and firewall rules which enables preventing all access to your database server until you specify which computers have permission to reach based on IP address of each request. In addition, Auditing feature of Azure SQL Database helps you to record database events to an audit log which allows you to understand, analyze and investigate previous database activities to identify potential threats and security violations. As it is mentioned earlier, thread detection alerts in case of injection attacks, suspicious activities and potential vulnerabilities. It is an extra feature that does not come automatically but it can be access from Azure Security Center. The cost of using thread detection feature is \$15 per server per month.



## d) Backup

Data is a very valuable asset to companies nowadays and for that reason it is important to guard it appropriately by creating backups. Azure has a backup Service, named *Azure Backup*, which automatically allocates and manages backup storage, and it uses a pay-as-you-use model meaning that the customer pays for the storage consumed. It also offers two backup possibilities:

- Locally redundant storage (LRS): Replicates the data storage three times within the same datacenter.
- Geo-redundant storage (GRS): replicates data to a secondary region (hundreds of miles away from the primary location of the source data). Its costlier than LRS but it provides more security in case the datacenter as a whole has a problem.

These backups are based on *Recovery Vaults* which are used to hold data such as backup copies, recovery points, and backup policies. These offers:

**Application-consistent backup:** Back-ups at a certain point in time have all the necessary data which means there is no need for additional fixes for restoration, making it faster.

**Unlimited data transfer:** Azure offers unlimited outbound data, data that is transferred from the Recovery vaults, as well as inbound data transfer as long as it is not excessively large.

**Long-term retention:** Recovery services can keep data for as long as the client wishes. They offer a limit of 9999 data recovery points per instance. A protected instance is a computer, server (physical or virtual), or workload configured to backup data. This means that if a daily recovery point was created, there would be a backup for 27 years.

**Incremental back-up:** Azure Backup components support incremental backups meaning that it updates only the changes made since the last backup. This makes recovery storage and time efficient. The next graph shows the differences between the different types of backup:

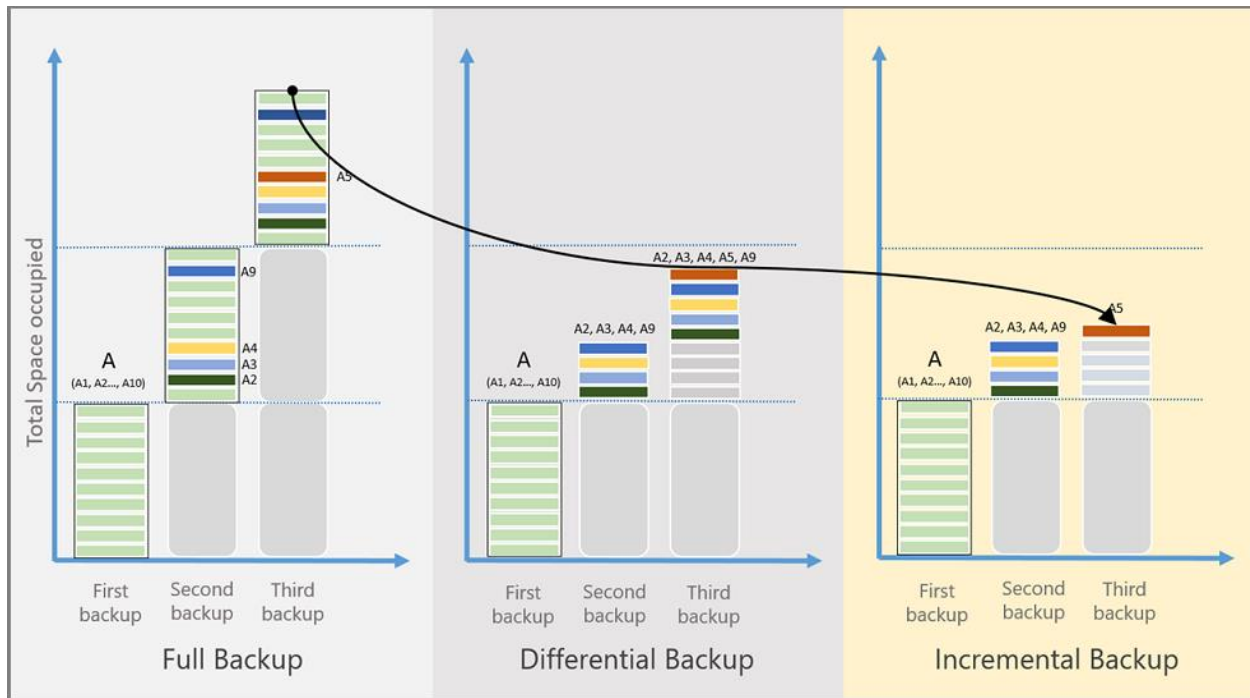


Figure 27: Azure incremental backup strategy

Premium Service VMs: The client might also decide to upgrade to premium backups which serves to protect solid-state drive (SSD)-based storage designed to support I/O-intensive workloads.

Azure Site Recovery: Apart from backups, Azure also offers Site Recovery which, in contrast with backups to a granular level, allows to restore workloads as it coordinates virtual-machine and physical-server replication, failover, and failback. Both services are needed for a full recuperation of the system.

All of these characteristics come at a price depending mainly on the backup possibilities discussed earlier and on the amount of storage needed to backup. Assuming a data size, average for a medium company, of 1000 TB for 10 Azure VMs (1 Petabyte), for 24 daily recovery points (one per hour), the cost calculator gives:

- Locally redundant storage (LRS): US 40.087,96/ month

**Azure Backup**

REGION: East US TYPE: Azure VMs

10 VMs × 100 TB = \$20,500.00

**Storage**

REDUNDANCY: LRS AVERAGE DAILY CHURN: Moderate

**Retention Range of RPs (Recovery Points)**

<b>24</b> Daily RPs (Days)	<b>0</b> Weekly RPs (Weeks)	<b>0</b> Monthly RPs (Months)	<b>0</b> Yearly RPs (Years)
----------------------------------	-----------------------------------	-------------------------------------	-----------------------------------

**Storage Cost Calculation**

Choose one of the following options to see your storage cost estimates at different points in time.

- After 1st month
- After 1st year
- At Steady State

**Total Storage** 1,720 TB = \$40,087.96 Per month

Figure 28: Azure Backup Calculator for LRS

- Geo-redundant storage (GRS): Will give double the price of LRS to US 80.175,92/month

**Azure Backup**

REGION: East US TYPE: Azure VMs

10 VMs × 100 TB = \$20,500.00

**Storage**

REDUNDANCY: GRS AVERAGE DAILY CHURN: Moderate

Retention Range of RPs (Recovery Points)

24 Daily RPs (Days) 0 Weekly RPs (Weeks) 0 Monthly RPs (Months) 0 Yearly RPs (Years)

Storage Cost Calculation

Choose one of the following options to see your storage cost estimates at different points in time.

- After 1st month
- After 1st year
- At Steady State

Total Storage **1,720 TB** = \$80,175.92 Per month

Figure 29: Azure Backup Calculator for GRS

This might seem an exaggeration but comparing to a local server it might be beneficial. When managing a self-ran back-up service, it is necessary to know that the hardware needed to guard the copies will almost be the same as the main data center itself. This will mean a huge investment on recovery servers, which will also be in need of upgrade and constant maintenance. This makes it a very costly investment with low flexibility as the database scales-up.

## 7) CONCLUSIONS

As a conclusion, even though traditional SQL Server might be less expensive (depending on different factors based on the defined structure chosen), it should be kept in mind that, it requires purchase of dedicated hardware, installation, configuration and maintenance overhead. In addition, change in requirement, system updates, licensing requires a significant investment of effort, money and time. Also, traditional SQL Server most likely needs to be renewed every five years which means that the whole process needs to be redone from the beginning.

Besides traditional SQL, Azure SQL Database provides high availability most of the time, contain backup files, recover in case of failure, secure your data and be capable of handling load balancing. In addition, it provides automation, less overhead and additional services that might be useful in the future such as data analysis and malware detection.

All of these, of course, depends on the client's requirements, because if it is a client of scale of a global massive corporation or it is an organization that is very static data-wise, it might be wiser to implement an on-premise data design.

## 8) REFERENCES

- [1] Columbus, L. (2017). *Roundup of Cloud Forecasts and Market Estimates, 2012* Retrieved from <https://www.cloudcomputing-news.net/news/2012/jan/18/roundup-of-cloud-computing-forecasts-and-market-estimates-2012/> [Accessed 29 Sep. 2017]
- [2] Joe0. (2017, September 29). Joe0. Retrieved from <https://www.joe0.com/2017/09/29/the-future-of-cloud-computing-relative-to-becoming-a-monopoly/>
- [3] Marinescu, D. C. (2017). *Cloud computing: Theory and practice*. Cambridge, MA: Elsevier. Cloud Databases – Amazon Web Services (AWS). (n.d.). Retrieved from <https://aws.amazon.com/products/databases/>
- [4] Pros and Cons of Database as a Service (DBaaS). (2018, August 15). Retrieved from <https://www.objectrocket.com/blog/company/advantages-and-disadvantages-of-dbaas/>
- [5] Adkinn. (n.d.). Browse learning. Retrieved from <https://docs.microsoft.com/en-us/learn/modules/>
- [6] Microsoft Azure Cloud Computing Platform & Services. (n.d.). Retrieved from <https://azure.microsoft.com/>
- [7] Beaulieu, Alan. *Learning SQL*. OReilly Media, 2009.
- [8] Cloud vs. On-Premises – Hard Dollar Costs. (n.d.). Retrieved from <https://www.linkedin.com/pulse/cloud-vs-on-premises-hard-dollar-costs-greg-deckler>
- [9] Coding Horror. (n.d.). Retrieved from <https://blog.codinghorror.com/scaling-up-vs-scaling-out-hidden-costs/>
- [10] SQL Azure Architecture. (2014, September 10). Retrieved from <https://www.sqlservergeeks.com/sql-server-sql-azure-architecture/>
- [11] Cuppett, Michael. *DeVops, DBAs, and DBaaS: Managing Data Platforms to Support Continuous Integration*. Apress., 2016.
- [12] Chao, Lee. *Cloud Database Development and Management*. CRC Press, 2014.
- [13] Ximenes, Filipe. “Blog.” *Understanding Database Multitenancy – Vinta Software*, 5 July 2016, [www.vinta.com.br/blog/2016/understanding-database-multitenancy/](http://www.vinta.com.br/blog/2016/understanding-database-multitenancy/)
- [14] “What the Heck Is a DTU?” *SQLPerformance.com*, 18 June 2018, <https://sqlperformance.com/2017/03/azure/what-the-heck-is-a-dtu>

[15] Rudy, Melissa. "Cloud vs. Local Backup: Which Do You Need?" *Techopedia.com*, 28 July 2017, <https://www.techopedia.com/2/29440/trends/cloud-computing/cloud-vs-local-backup-which-do-you-need>

[16] Crabtree, Joe. "10 Best Microsoft Azure Features For 2015." *Perficient Blogs*, Perficient Blogs, 8 Jan. 2015, <https://blogs.perficient.com/2015/01/07/10-best-microsoft-azure-features-for-2015/>

[16] E. Gelogo, Y.S. Lee. *Database management system as a cloud service*. International Journal of Future Generation Communication and Networking (2012), p. 5

[17] Ylber Januzaj; Jaumin Ajdari; Besnik Selimi. *DBMS as a Cloud service: Advantages and Disadvantages*. Procedia - Social and Behavioral Sciences, ISSN: 1877-0428, Vol: 195, Page: 1851-1859. Elsevier BV., 2015

[18] Mehak, Faria & Masood, Rahat & Ghazi, Yumna & Shibli, Awais & Khan, Sharifullah. (2014). *Security Aspects of Database-as-a-Service (DBaaS) in Cloud Computing*. 10.1007/978-3-319-10530-7\_13

[19] w3resource. *SQL Sample Database*. <https://www.w3resource.com/sql/sql-table.php>